特集

開発プロセスの標準化とWebアプリケーション開発への対応

Standardization of Development Process, and Additional Efforts Focusing on Web Application Development

あらまし

情報化が著しい現在，情報システムは企業活動を営む上で必要不可欠なビジネス基盤となっている。この情報システム開発においては，機能の多様化，作業の専業化や分業化などにより，関係者は多岐にわたるため，お互いの共通認識の下で役割分担や責任範囲を明確にして協力し合い作業を遂行する必要がある。

本稿では，企画から運用・保守に至るライフサイクル全般の作業を網羅し，分かりやすく体系化したシステム開発の地図「SDEM21」を紹介し，つぎに様々なWebアプリケーションの実装技術に柔軟に対応し，プロジェクトに適した開発標準を容易に構築することができる「ComponentAA開発標準」を紹介する。最後に，Webアプリケーションのセキュリティ対策にかかわる標準化として「セキュアWebアプリケーション」を紹介する。

Abstract

In this time of remarkable informationization, information systems have become an essential business base for managing enterprise activities. Information system workers cover a lot of ground by diversifying the functions of systems, and tasks should be accomplished through the division and specialization of labor. It is therefore necessary to clarify the roles and ranges of responsibility of workers and for workers to cooperate with each other. To meet this need, Fujitsu has developed the SDEM21 system development map, which plainly systematizes the entire life cycle of a task from planning to operation and maintenance.

Web application development has accelerated in recent years, and to assist in this area, Fujitsu has developed the ComponentAA Development Method. This method makes it easy to construct a flexible project development standard for various technologies for packaging Web applications. Moreover, Fujitsu has developed Secure Web application as a standard for security countermeasures of Web application. This provides customers with stable and secure Web services.

齊藤涼子（さいとう　りょうこ）

SDAS推進統括部SDEM推進部 所属
現在，社内の標準プロセスの整備に従事。

沖山　智（おきやま　さとる）

SDAS推進統括部SDAS技術部 所属
現在，Webアプリケーションの開発技術の整備に従事。

平井　宣（ひらい　ふさみ）

SDAS推進統括部SDAS技術部 所属
現在，SDASにかかわる開発技術の整備に従事。

ま え が き

情報化が著しい昨今では，情報システムは企業活動を営む上で必要不可欠なビジネス基盤となっている。そのため，トップ，利用部門も含めた多くの関係者が関与するようになった。また，システムにおけるセキュリティ対策の重要性が増している。

本稿では，富士通におけるシステム開発の進め方の標準について「SDEM21（Solution-oriented system Development Engineering Methodology 21）」を紹介し，さらに，現在のシステム形態の主流であるWebアプリケーション開発の観点も含めながら「ComponentAA開発標準」「セキュアWebアプリケーション」を紹介する。

SDEM21

本章では，SDEM21の概要とSDEM21をより実践的に使うためのコンテンツについて説明する。

● SDEM21の概要

システム開発においては，機能の多様化，作業の専業化や分業化などにより，関係者は多岐にわたる。システムの企画，開発，運用・保守を成功裏に成し遂げるためには，これらの人々が同じ言葉でコミュニケーションを取りながら，お互いの共通認識の下，役割分担や責任範囲を明確にして，協力し合い作業を遂行することが不可欠である。

そこで，SDEM21では，企画から運用・保守に至るライフサイクル全般の作業を網羅し，分かりやすく体系化している。いわば，システム開発を可視化する。現場SE中心のWG活動や，富士通の幾多のプロジェクト経験に基づいて開発したもので，ISO国際規格へも対応している。

以下，SDEM21の「作業体系」，「工程（品質保証のV字モデル）」，「カテゴリ」を順に説明していく。

【作業体系】

設計，製造，テストといった工程を横軸に，業務，インフラ，運用・移行といった開発を遂行するための作業の分類であるカテゴリを縦軸に配置し，その横軸と縦軸の交差する各点にやるべき作業（WBS）を定めている（**図**-1）。

このマトリックスによって，多岐にわたる作業の全体像を，多くの関係者に理解しやすいように示すことで，網羅性を確保することができる。

【工程（品質保証のV字モデル）】

システムを企画，開発し，運用・保守するという連続した作業の流れの中で，その過程の管理をするために，工程を定義している（**表**-1）。とくに，開発プロセスでは，工程を品質保証のＶ字モデルに基づいて設定している。

品質保証のV字モデルは，**図**-2のように開発対象をシステム構成要素（業務→システム→プロセス→プログラム）に分解していく過程と，統合の過程を対応させることで，品質を保証していくモデルである。設計とテストを対にすることで，設計に応じたテストでの検証・妥当性確認の内容が明確になり，より高い品質を作り込むことができる。もちろん，各設計段階でのレビューやプロトタイプなどによる検証・妥当性確認や各テスト段階での品質判断は必要不可欠である。

【カテゴリ】

カテゴリとは，各プロセスを遂行するために必要な作業の種類の大分類であり，作業の関連性，必要な知識，技術の関連性をもとに分類したものである。企画，開発，運用・保守に携わる関係者の間での役割分担，体制作りのベースとして活用できる。

カテゴリの例を挙げると，開発プロセスでは，業務，業務システム仕様，アプリケーション，インフラ，運用・移行（セキュリティ含む），開発支援，プロジェクト管理である。

このように，作業の種類を明確に分類することで，作業・見積り・分担漏れを防止するとともに，進捗

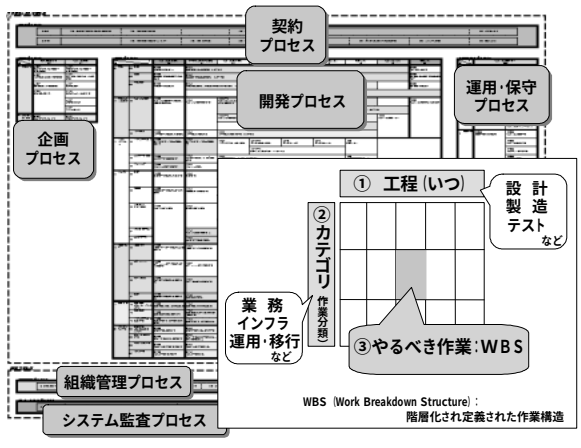

図-1　SDEM21作業体系
Fig.1-Outline of SDEM21.

表-1　工程の定義

企画		情報化構想立案	VP	・情報化戦略を策定し，実行の優先順位を決め，中長期計画を策定。
		システム企画	SP	・業務システムの現状分析と要件定義を行い，投資効果を評価し，開発の意思決定。
開発	設計	システム方式設計	SA	・システム化に対する要件を確認し，業務分析によりシステム化の範囲を確定。 ・システム方式設計と実現性の検討を行い，プロジェクト開発計画を立案し，プロジェクト運営のための管理方法を準備。
		ユーザインタフェース設計	UI	・業務システム仕様（プロセス機能，データ構造，画面，帳票）を設計。 ・システム方式設計の詳細化，および運用・移行方法を設計。 ・全体テスト計画を立案。
		システム構造設計	SS	・プロセスをプログラムに分割し，システムの内部構造を確定し,共通プログラムを設計。 ・運用管理システム，セキュリティシステム，移行ツールを設計。 ・システムテスト計画，運用テスト計画を立案。
	製造	プログラム構造設計	PS	・プログラムの構造を決め，ロジックを定義。
		プログラミング	PG	・プログラム構造設計に従ってプログラムを作成し，動作を確認。
		プログラムテスト	PT	・プログラムテスト仕様に従ってテストを実施し，品質を検証。
	テスト	結合テスト	IT	・プログラムを結合して，プロセス単位のテストを実施し，品質を検証。 ・外部システムとのインタフェースを含むすべてのプロセス間のインタフェーステストを実施。 ・必要に応じて結合レベルや処理形態例（オン/バッチ）でIT工程を分割。
		システムテスト	ST	・実機上で業務システム機能をテスト。 ・性能，信頼性，運用性，セキュリティなど，システム全体の検証を実施。
		運用テスト	OT	・実機，実環境，本番データで，利用者による仮運用を実施。 ・業務システム機能，性能，信頼性，運用性，セキュリティなどの妥当性を確認。 ・本稼働への移行の意思決定を行い，業務を移行。
運用・保守		システム運用・保守	OM	・業務運用管理，利用部門支援，システム運用，運用管理，システム保守の方法を確認。 ・トラブル・Q/A対応，ネットワーク，ハードウェア，ソフトウェア，アプリケーション使用などの変更管理を実施し，改善計画を立案。

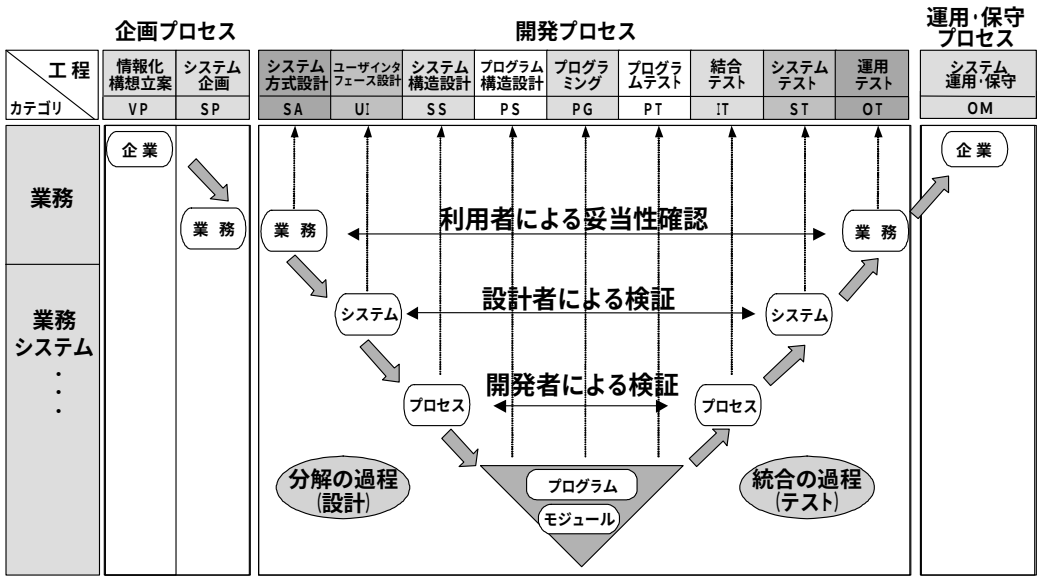

図-2　品質保証のV字モデル
Fig.2-V model for quality assurance.

管理も可能となる。また，明確にした作業分類ごとに作業量を見積り，品質目標を定め，実績を管理することで，進捗度合が測定しやすくなる。さらに，体系化された作業は，人材育成（若年層への教育，人材像の定義への活用など）やプロセス改善（改善点を見つけるための評価対象としての活用など）のベースになる。

● SDEMをより実践的に使うためのコンテンツ

現在，プロジェクトの開発期間は短期間化している。プロジェクトの立上げに多くの時間を割けない。そこで，SEがアクションに結び付けるべき行動規範や，プロジェクトがなるべく少ない労力で活用で

きる開発標準を整備し，SEに提供している。

【SDEM実践テンプレート】

これはSDEMのマトリックスをベースに，プロジェクトの遂行に必要なテーマごと（プロジェクト計画，業務基盤，システム基盤など，計16テーマ）に，作業にどういう姿勢で取り組むか，結果をもとにどうアクションをとるか，といったポイント（三つの行動規範）や開発作業上のノウハウをコンパクトにまとめたものである。

【SDEM実践開発標準】

この開発標準は，WBSをPERTで手順化・具体化し，ドキュメントのひな型を兼ね備えた実践的な開発標準であり，業務アプリケーション開発を対象とした「業務アプリ開発一般型」（非Webアプリケーション，COBOL/C，従来表記）や「ComponentAA開発標準」{Webアプリケーション，Java，UML（Unified Modeling Language）表記}，また，システム基盤開発を対象とした「基盤作業キット」を用意している。

ComponentAA開発標準

ComponentAA開発標準[1]はWebアプリケーションの様々な技術に対応した一貫性のある開発標準の提供を目的としており，開発作業手順標準/ドキュメント標準，および各種の開発技法などから構成される（**図**-3）。以下では，ComponentAA開発標準がWebアプリケーションの技術にどのように対応しているかを概説する。

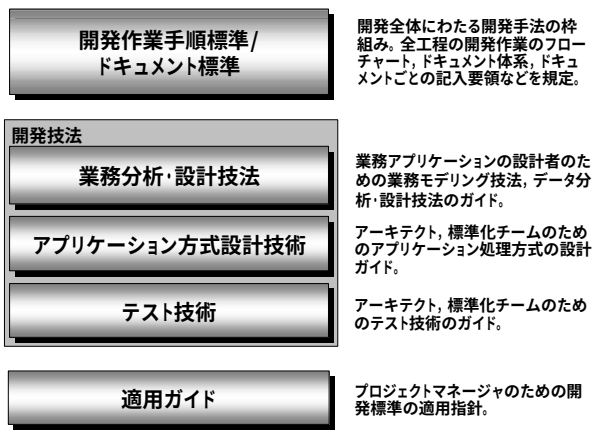

図-3　ComponentAA開発標準の技術体系
Fig.3-Technology system of ComponentAA Development Method.

● Webアプリケーションの実装技術への対応

メインフレームやクライアント/サーバシステム用の開発標準をWebアプリケーションの開発に適用することは難しい。なぜなら，前提となる実装技術（ミドルウェアやフレームワークなど）の違いによりアプリケーションの構造が異なり，設計項目や実装対象が全く異なっているためである。もし無理に適用を試みようとすると，開発標準の大幅な修正が必要になる。また，Webアプリケーションの実装技術には複数の選択肢が存在する。例えば，画面はServlet/JSP技術を使うことも，Applet技術を使うこともできる。そのため，選択された技術によって設計項目の変動が生じる。このこともWebアプリケーションの開発標準の構築を難しくしている。

ComponentAA開発標準は，Webアプリケーションの開発に適した3層コンポーネント構造[2]（画面の表示や制御を行うプレゼンテーション層，業務ロジックを実行するコントロール層，およびデータベースへのアクセスやデータの一貫性維持に責任を持つモデル層で構成するアプリケーションの論理的構造）を基本とし，各層のコンポーネントの設計手順や設計ドキュメントを規定している（**図**-4）。また，選択可能な実装技術ごとに設計のガイドが提供されている。例えば，プレゼンテーション層にServlet/JSP技術を適用する場合はServlet/JSP用の設計ガイドを用い，Applet技術を用いる場合はApplet用の設計ガイドを用いればよい。これらの規定や設計ガイドを組み合わせることで，様々なWebアプリケーションの実装技術に柔軟に対応し，

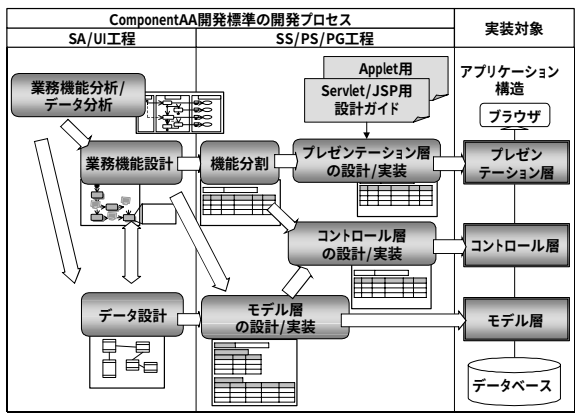

図-4　Webアプリケーションの構造に対応した開発プロセス
Fig.4-Development process corresponding to Web application architecture.

プロジェクトに適した開発標準を容易に構築することができる。

● UMLの活用

UML[3],[4]はオブジェクト指向分析・設計のための汎用的な表記法であり，広く普及してきている。しかし，UMLの適用に当たっては具体的な開発手順の中で使用目的や使用方法を明確に規定しなければ有効に活用することはできない。例えば，「UMLのクラス図を使ってクラス設計せよ」と言っただけでは，出来上がる設計書の内容や品質はばらばらになってしまい，効率的な開発はできない。

ComponentAA開発標準ではUMLの使用目的や使用方法を具体的に提示している。例えば，「画面設計担当者は，ユーザインタフェース設計工程で，画面の遷移を設計するために，画面遷移をUMLのアクティビティ図で記述する」というように，誰が，いつ，何のために，何をUMLで記述するかを規定している。このようにComponentAA開発標準と組み合わせて用いることで，UMLを有効に活用することができる。

● 実績のあるデータ分析・設計技法の統合

ビジネス要件を適切にシステム化するためには，データ分析・設計技法が有効である。このことは，オブジェクト指向技術に基づいて実装を行うWebアプリケーションの開発の場合でも同じである。ここで必要になることは，上流工程でのエンティティ関連図（ER図）などのデータ分析・設計と，オブジェクト指向技術に基づく実装の間のギャップを埋めることである。オブジェクト指向では，設計や実装の自由度は大きいため，適切な設計指針が存在しなければアプリケーション構造が不統一となり，品質や保守性が著しく低下する。

ComponentAA開発標準では，データ分析・設計をモデル層のコンポーネントの実装に反映する設計指針を提供しており，データ分析・設計技法とオブジェクト指向の実装作業を矛盾なく一つの開発標準の中に統合している。

● 実績と今後の展開

ComponentAA開発標準は，2004年4月のリリース以来，50以上のプロジェクトに適用されている。また，ドキュメント標準の部分をWebで一般公開しており，1,080名以上の方がダウンロードし利用されている。

Webアプリケーションの技術は発展を続けており，現在でも多くの実装技術やフレームワークが次々と登場し，いくつかは普及の兆しを見せている。このような中で，WebアプリケーションをターゲットとするcomponentAA開発標準は，今後も新たな技術に対応し，最適な開発標準の提供を続けていく予定である。

セキュアWebアプリケーション

Webアプリケーション開発の加速と同時に，インジェクション攻撃やフィッシングなど，インターネットなど外部からの攻撃が増加している。例として，ぜい弱性を利用したSQLインジェクション攻撃を**図-5**に示す。この例では，購買情報照会画面の商品コードに意図しない値「A0012' OR 'A'='A」が入力され，Webアプリケーションが全文検索のSQL文として解釈してしまい，その結果公開前の取引情報や別の取引先別単価などの取引情報が漏えいしてしまう。それにより，高品質，高信頼な情報システムを構築したにもかかわらず，アプリケーションのぜい弱性が検出され，改修にかかわる膨大なコストの工面やサービス停止を招くことで信用問題への発展も有り得る。

このような問題から，ぜい弱性対策は，Webアプリケーション開発に取り組む上で重要な位置付けとなりつつある[5],[6]。

本章では，富士通が取り組んでいるSDASセキュアプログラミングについて説明する。

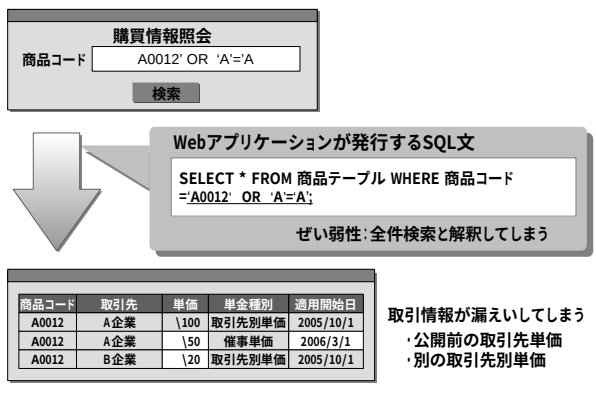

図-5 ぜい弱性を利用したSQLインジェクション攻撃の例
Fig.5-Examples for SQL injection using vulnerability of Web applications.

●ぜい弱性対策の活性化

現在よく見られる対策は，有償化サービスを別途適用し品質を確保するスタイルが主流である。SecurityFocusが運営しているBugTraqメーリングリストやWebアプリケーションセキュリティの啓蒙活動を行っているOWASP（The Open Web Application Security Project）[7]などは，この分野の情報ソースが指摘しているぜい弱性情報を分析し，診断ツールを駆使することによって，対処方法を提供してきた。

国内では，2004年7月に経済産業省より「ソフトウエア等脆弱性関連情報取扱基準」が告示され，IPAセキュリティセンター（IPA/ISEC）がWebアプリケーションのぜい弱性関連情報の受付機関として，届出の受付を開始した[8] また，ビジネスとしてWebアプリケーションのぜい弱性診断サービスも開始されている。

●　ぜい弱性対策における課題

これらのサービスはWebアプリケーション開発の技術変化に合わせて多様化するぜい弱性を抑え，どうやれば見つかるかのノウハウを駆使して脅威分析や対処方法を提供するため，お客様のシステム形態に合わせたきめ細かい対策ができる反面，どうしてもコスト高になってしまう。

また，これらのサービスはアプリケーションが稼働する直前に適用されることが多く，開発過程で対策を講じるプロセスはまだ十分には確立されていない。

これを確立するためには，開発者の役割分担に応じた工夫が必要で，業務仕様に基づいて業務ロジックを開発するチームと，その業務ロジックがWebアプリケーションとして機能する処理方式を提供するチームのそれぞれの視点で対策を講じることが重要である。そもそもセキュリティ対策はチーム横断的な要素であり，徹底という意味で誰に何をやってもらうかということを考えないと対策手順の定型化ができないため，コスト増や診断の長期化に発展してしまう。

●　対策の型決め

図-6に示すように，SDASでは開発標準の一環でセキュアなWebアプリケーションを開発するためのガイド「セキュアWebプログラミングガイド」を社内提供している。アプリケーション開発コストに含めるべき対策（当たり前の領域）と非機能要件の実現（信頼性や運用対策などと同様に不可欠な要件としてきちんと考える）とを分けて考え，事前にお客様とアプリケーション開発コストに含めるべき項目を決められるようになっている。このベースラインの確立と既知のぜい弱性対策をあらかじめ体系化しておくことで，日々追加される可能性のあるぜい弱性対策と区別でき，ベースラインとしての対策なのか，コストを投じて対処すべきものなのかの切分けも可能になる。

また，やってはいけないことを，いつ，どのようにチェックするかの工夫が重要となる。これまでは，その検出を第三者が代行することで，コスト高となっていた。このコストを下げるために，以下の3点が重要となる（**図**-7）。

（1） みずからが実施・是正するための確認方法の提供

（2） 量産するものについては，検出するためのツールを提供することが必要

（3） 上記（2）を実現するための事前検証されたフレームワーク，アプリケーション処理方式の適用

●　さらなる型決めに向けて

対策の型決めによるコストダウンはまだこれからの取組みでもあり，全社的な普及を考えた場合，事前問診や診断手順の改善を，実践を通して更に進めていく。

また，ぜい弱性対策の原理・原則は比較的安定しているものの，技術変化に伴う診断方法や処理方式の検討などは，スピードが要求される。また脅威分

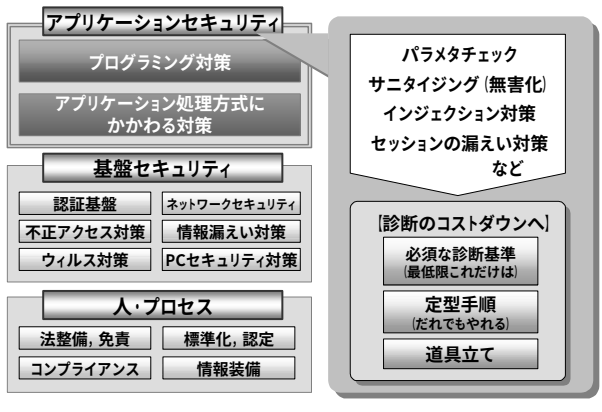

図-6　Webアプリケーションセキュリティの位置付け
Fig.6-Scope of SDAS Web application securities.

カテゴリ		対策数	道具立て
業務ロジック	AppletやJavaコード	35	SIMPLIA/JF Kiyackerなどの規約チェッカ
	Servlet (JSPやJavaScript)	11	文字列検索やAppScanなどの診断ツール
処理方式	・認証などのアプリケーション処理方式 ・開発標準など	16	コンサルティング (フレームワークがカバーする範囲を明確化)
運用・環境など		13	

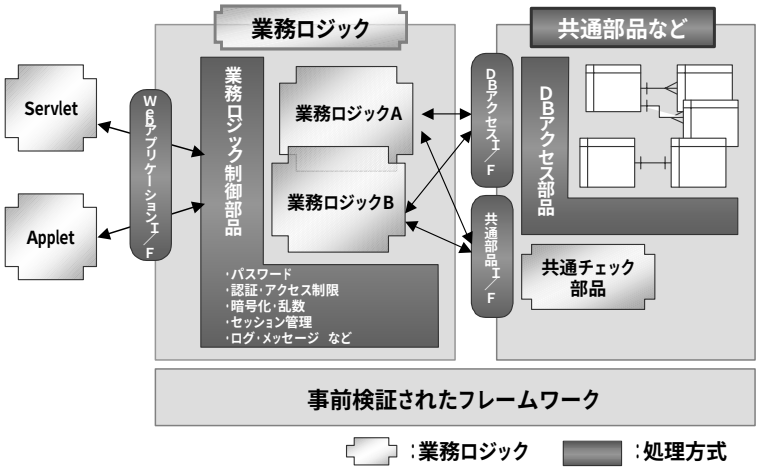

図-7　Webアプリケーションのぜい弱性対策における型決め
Fig.7-Protection profiles against vulnerabilities of Web applications.

析を伴うような認証，暗号化対策などベースラインで済まない場合の判断や実績のある処理方式・ノウハウの蓄積，情報共有を軌道に乗せたい。

む　す　び

本稿では，システム構築の基本的な考え方である「SDEM21」およびSEの行動規範を示した「SDEM実践テンプレート」，プロジェクトの現場の標準化作業に活用できる実践的/具体的な開発標準「SDEM実践開発標準」「ComponentAA開発標準」「セキュアWebプログラミングガイド」を紹介した。今後も，プロジェクトの確実な推進に役立つように，新技術の動向も踏まえながら，継続的に普及・強化していく予定である。

参 考 文 献

(1)　富士通：ComponentAA開発標準.
http://segroup.fujitsu.com/sdas/technology/develop_guide/1_caa.html

(2)　EJBコンポーネントに関するコンソーシアム：再利用可能EJBコンポーネントデザイン（テクニカルレポート）DE-00-01.
http://www.ejbcons.gr.jp/rules/DE-00-01-PR.PDF

(3)　Object Management Group ： Superstructure Specification, v2.0 (formal/05-07-04UML).
http://www.omg.org/

(4)　Object Management Group, OMG Japan SIG翻訳委員会UML作業部会訳：UML仕様書．アスキー，2001.

(5)　高木浩光：安全なWebアプリ開発40箇条の鉄則. 2003.
http://java-house.jp/~takagi/paper/idg-jwd2003-takagi-dist.pdf

(6)　IPA ISEC：セキュアプログラミング講座.
http://www.ipa.go.jp/security/awareness/vendor/programming/

(7)　OWASP：Webアプリのぜい弱性ワースト10.
http://www.owasp.org/

(8)　早貸淳子：脆弱性情報の取り扱いについて－情報セキュリティ早期警戒パートナーシップの概要と運用の状況－．情報処理，Vol.46，No.6，p.662-671（2005）.